

Занятие 8

Алгоритмическая система Тьюринга

Первоначально, исходя из условия задачи, надо определить входной алфавит, т.е. символы, которые будут находиться на ленте для данной задачи. В процессе решения задачи – написания алгоритма для работы МТ входной алфавит может быть дополнен.

Затем придумать идею алгоритма, т.е. как будет двигаться головка МТ – управляющая головка (УГ), где будет начальное ее положение. Алгоритм работы МТ может немного напоминать алгоритм сортировки (надеюсь, что не запутала вас). Начальное положение головки МТ вы выбираете сами, исходя из идеи алгоритма. Так как лента бесконечная, то для конкретной задачи, для упрощения ее решения можно ограничить последовательность символов некоторыми специальными, либо ввести в рассмотрение знак пробела.

Определить начальное состояние. В большинстве случаев, это первое из состояний, записанных в таблице.

Получается, что в соответствии с алгоритмом УГ бежит направо – налево вдоль ленты и производит предписанные алгоритмом действия.

Для упрощения построения алгоритма можно нарисовать частный случай задачи и на нем попытаться придумать алгоритм, а потом, когда станет ясна идея, переделать алгоритм для любого варианта расположения символов на ленте для данной задачи.

Задача 1.

Построить МТ, которая определяет четность или нечетность числа 1 в строке. Конец последовательности помечается символом В, затем в эту ячейку будет записан результат.

Допущение: Управляющая головка (УГ) находится под первым символом последовательности (если специально не оговорено, определяется человеком решающим задачу).

			1	0	1	1	0	1	1	В		
^												

Алгоритмическая идея: МТ требуется 2 состояния: одно для нечетного числа 1, другое для четного. Исходя из условия, результат будет записан в ячейку с символом В – 0 при четном числе 1, 1 – при нечетном. Для учета уже посчитанных/пройденных 1, каждая встретившаяся и учтенная 1 стирается, т.е. на ее место записывается 0.

	S1	S2
0	S1 П	S2 П
1	0 S2 П	0 S1 П
В	0 S1 !	1 S2 !

Установим: S1 – состояние для четного числа 1, S2 – для нечетного.

В начальный момент времени, находимся в состоянии S1

Действие: УГ устанавливается в очередную ячейку. Если там записана 1, то на ее место записывается 0, МТ переходит в противоположное состояние, и УГ передвигается в следующую ячейку. Движение УГ слева направо. Если в рассматриваемой ячейке записан 0, то состояние не меняется, УГ передвигается вправо к следующей ячейке. Если встретился символ В – надо анализировать, в каком состоянии находится МТ в этот момент. Если в S1, то на место В записывается 0, если в S2, то записывается 1 и происходит остановка МТ.

Задача 2.

Построить МТ, для проверки скобочных выражений. МТ должна решить, является ли последовательность из левых и правых скобок правильной, т.е. каждой левой скобке (должна соответствовать правая). Начало и конец последовательности ограничены символами А.

Допущение: Управляющая головка (УГ) находится под первой слева скобкой.

	0	0	А	(()	())	А		
				Λ								

Алгоритмическая идея: Ищется первая правая) скобка, затем первая левая (, ей парная, и обе заменяются символом Х. Вычеркивание парных скобок продолжается до тех пор, пока не произойдет одно из событий:

- 1) МТ, продвигаясь влево не находит парного символа (, при достижении символа А она на его месте печатает символ 0 и останавливается.
- 2) МТ, продвигаясь вправо не находит ни одного символа) и достигает символа А. В этом случае МТ начинает движение влево и проверяет – не остался ли какой-то из символов (:
 - Если такой (символ найден, то на месте А печатается 0
 - Если нет символа (, то на месте символа А печатается 1.

Состояние S1 предписывает движение вправо (П)

Состояние S2 предписывает движение влево (Л)

Состояние S3 предписывает движение влево (Л), когда не найдено ни одной ")". Т.е. головка приходит к правому А и при продвижении влево при нахождении "(" переходит в S4, для того, чтобы поставить символ 0 вместо левого А

Начальное состояние S1.

	S1	S2	S3	S4
)	X S2 Л	Л	-	-
(П	X S1 П	S4 Л	Л
А	S3 Л	0 !	1 !	0 !
Х	П	Л	Л	Л

Проверк "-" в программе МТ означает, что быть такой ситуации для данной задачи не может.

Задача 3. Построить МТ, переворачивающую любое слово в алфавите $A=\{a,v\}$. Т.е. построить зеркальное отображение заданного слова.

Например.

Чтобы знать, где начинается слово, в соответствующую ячейку ленты запишем *. Конец последовательности символов слова означает пробел (\emptyset).

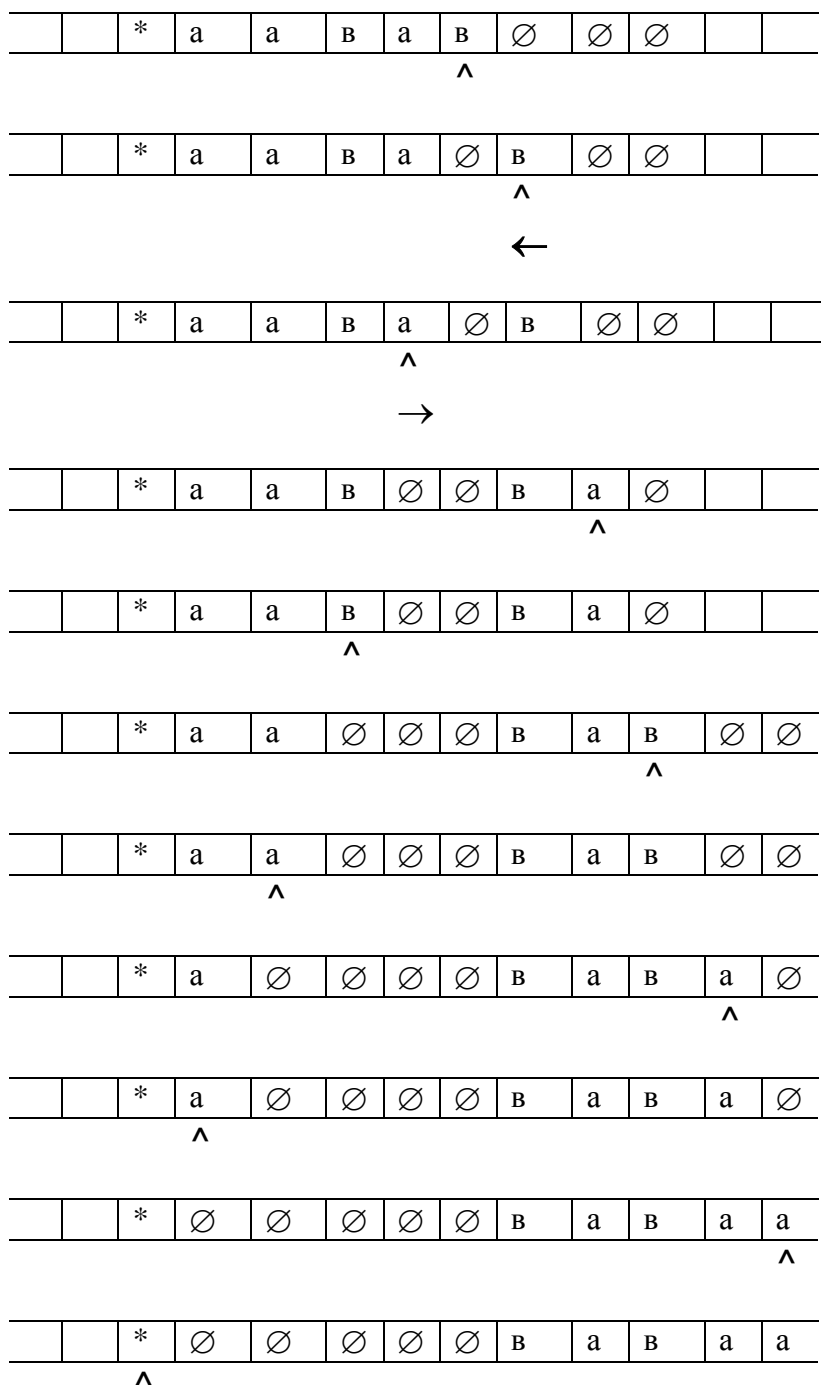
	*	а	а	в	а	в	\emptyset	\emptyset	\emptyset		
Λ											

Таким образом, алфавит для написания программы МТ будет состоять из: а, в, *, \emptyset .

Алгоритмическая идея: УГ устанавливается на последний символ слова. МТ находится в состоянии S1. Если это символ алфавита $A=\{a,v\}$, то символ стирается, т.е. вместо него ставится пробел \emptyset , МТ переходит в другое состояние, УГ начинает движение направо и ищет первый пробел. Найдя его она печатает на его месте стертый символ и переходит в состояние, отвечающее за продвижение налево, т.е. за возврат к анализируемому слову. В этом состоянии УГ движется налево до первого пробела, а найдя его сдвигается еще раз

налево (к следующей ячейке) и переходит в состояние S1, отвечающее нахождение очередного символа слова для стирания. Если в состоянии S1 МТ в ячейке обнаруживает *, то это означает, что все символы проанализированы, зеркальное отображение построено. Происходит останов МТ.

Пошаговый пример построения зеркального отображения слова.



Программа МТ

Начальное состояние S1.

	S1	S2	S3	S4
а	∅ S2 П	П	П	Л
в	∅ S3 П	П	П	Л
*	!	-	-	-
∅	Л	а S4 Л	в S4 Л	S1 Л

Пример 1. Алфавит – {a, b}.

Правила подстановки:

1. $aa \rightarrow a$

2. $bb \rightarrow b$

Исходное слово: bbbbaabb

1) bbbbaabb (1) bbbbaabb

2) bbbbaabb (2) bbbabb

3) bbbabb (2) bbabb

4) bbabb (2) babb

5) babb (2) bab

Итоговое слово - bab.

Последовательность шагов Вы видите на экране.

Алгоритм преобразования слова.

Согласно концепции нормальных алгоритмов Маркова преобразование слова P по правилам подстановки s_1, s_2, \dots, s_n представляет собой последовательность однотипных шагов.

Один шаг преобразования.

1) Последовательно просматриваются правила s_1, s_2 и т.д.

2) Если левая часть u_i некоторого правила s_i входит в слово, то вхождение u_i в слово заменяется правой частью правила v_i .

Данные шаги преобразования последовательно выполняются один за другим.